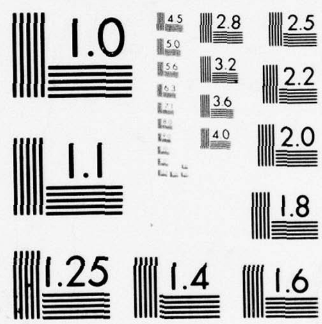
AD-A049 467 WASHINGTON UNIV SEATTLE DEPT OF MATHEMATICS F/G 12/2
USE OF FLOYD'S ALGORITHM TO FIND SHORTEST RESTRICTED PATHS.(U)
SEP 77 V KLEE, D LARMAN N00014-67-A-0103-0003
UNCLASSIFIED TR-62 NL

1 OF 1
AD
A049467

END
DATE
FILMED
3-78
DDC



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

AD A049467

AD No.
DDC FILE COPY

(12)

B.S

(6)

USE OF FLOYD'S ALGORITHM TO FIND SHORTEST RESTRICTED PATHS,

by

(10)

Victor Klee ~~and~~ David Larman

(9) Technical Report No. (14) TR-62

(11) September 1977

(12) 24p.

(15) Contract N00014-67-A-0103-0003

Project Number NR044 353

ACCESSION for		
NTIS	White Section	X
DDC	Buff Section	☐
UNANNOUNCED		☐
JUSTIFICATION........		
BY........		
DISTRIBUTION/AVAILABILITY CODES		
Dist.	AVAIL. and/or SPECIAL	
A		

Department of Mathematics
University of Washington
Seattle, Washington 98195

This research was supported in part by the Office of Naval Research.

370 275

JB

USE OF FLOYD'S ALGORITHM TO FIND SHORTEST RESTRICTED PATHS

VICTOR KLEE and DAVID LARMAN

University of Washington, Seattle and University College, London

Abstract. In a directed network with no negative circuit, Floyd's algorithm finds, for each pair of nodes x and y, a shortest path from x to y. Here the procedure is extended to minimize more general length-functions over sets of paths that are restricted in various ways.

Introduction.

Throughout this paper, G denotes a complete directed graph with n nodes and $n^2 - n$ edges. Each edge is an ordered pair (i, j) of nodes and has as its length a number $\lambda(i, j) \in R^* =]-\infty, \infty]$. For notational convenience G's node-set is assumed to be the set $N = \{1, \cdots, n\}$.

A walk from x to y is a node-sequence $(x_0, \cdots, x_t)$ such that $x_0 = x$, $x_t = y$, and $t > 0$. It is a chain if no node is repeated and a circuit if $x_t = x_0$ but there is otherwise no repetition. Both chains and circuits are called paths, a practice that is unusual but is convenient for our purposes.

Floyd's algorithm [R][W][F][H2][L] initializes $S[i, j] \leftarrow \lambda(i, j)$ for all $i, j \in N$ and then proceeds as follows:

```
for k ← 1 until n do
  for i ← 1 until n do
    for j ← 1 until n do
      S[i, j] ← min{S[i, j], S[i, k] + S[k, j]}.
```

If there are no circuits of negative length then $S[x, y]$ emerges as the length of a shortest path from x to y. The computation is easily modified to find shortest paths in addition to their lengths.

In the present paper the procedure is extended to deal with a family $\underline{\underline{F}}$ of sets of walks and with a walk-length function L more general than the usual one. Under suitable assumptions the extended procedure finds, for each choice of $x, y \in N$ and $\underset{\sim}{Z} \in \underline{\underline{F}}$, a shortest $\underset{\sim}{Z}$-path from x to y. That is, L is minimized over the set of all paths from x to y that belong to $\underset{\sim}{Z}$. In the "classical" case, $\underset{\sim}{Z}$ is the set of all paths (or walks) in G, $\underline{\underline{F}} = \{\underset{\sim}{Z}\}$, and the length of a walk is the sum of the length of its edges.

The Assumptions.

The function L is used to measure the length of a walk in terms of the lengths of its edges. It is assumed the range of L is contained in R^*, the domain of L is the set of all finite sequences in R^*, and the following two conditions are satisfied:

(1) if $\alpha_1, \cdots, \alpha_t \in R^*$ and $0<s<t$ then

$$L(\alpha_1, \cdots, \alpha_t) = L(L(\alpha_1, \cdots, \alpha_s), L(\alpha_{s+1}, \cdots, \alpha_t))$$

(2) if $\alpha_1, \alpha_2, \beta_1, \beta_2 \in \text{rng } L$ with $\alpha_1 \leq \beta_1$ and $\alpha_2 \leq \beta_2$ then $L(\alpha_1, \alpha_2) \leq L(\beta_1, \beta_2)$.

The length of a walk $W = (x_0, \cdots, x_t)$ is defined as

$$L_\lambda(W) = L(\lambda(x_0, x_1), \cdots, \lambda(x_{t-1}, x_t)).$$

By (1), $L_\lambda(UV) = L_\lambda(L_\lambda(U), L_\lambda(V))$ when U is a walk from x to y and V is a walk from y to z. Here UV denotes the walk that follows U from x to y and then follows V from y to z.

Among the admissible functions L are

$$L_p(\alpha_1, \cdots, \alpha_t) = (\alpha_1^p + \cdots + \alpha_t^p)^{1/p} \text{ for an integer } p > 0,$$

$$L_\infty(\alpha_1, \cdots, \alpha_t) = \max(\alpha_1, \cdots, \alpha_t),$$

$$L^p(\alpha_1, \cdots, \alpha_t) = (|\alpha_1|^p + \cdots + |\alpha_t|^p)^{1/p} \text{ for a real } p > 0.$$

The usual L is L_1. The function L_∞ is also of practical interest, for if G is initially equipped with nonnegative real edge-weights $\gamma(i, j)$ representing flow capacities and if $\lambda(i, j) = -\gamma(i, j)$ for all i and j, then the shortest paths with respect to L_∞ are those of maximum flow capacity for specified initial and terminal nodes [H1][H2][L].

When $\underset{\sim}{U}$ and $\underset{\sim}{V}$ are sets of walks let

$$\underset{\sim}{U}\underset{\sim}{V} = \{UV\colon U\in\underset{\sim}{U},\ V\in\underset{\sim}{V},\ U \text{ ends where } V \text{ starts}\}.$$

Thus a walk $(x_0,\cdots, x_t)$ belongs to $\underset{\sim}{U}\underset{\sim}{V}$ if and only if there exists s such that $0<s<t$, $(x_0,\cdots, x_s)\in\underset{\sim}{U}$ and $(x_s,\cdots, x_t)\in\underset{\sim}{V}$. The first assumption about $\underline{\underline{F}}$ is:

(3) *if* $0<s<t$ *and* $(x_0,\cdots, x_t)\in\underset{\sim}{W}\in\underline{\underline{F}}$ *then there exist* $\underset{\sim}{U}$, $\underset{\sim}{V}$ *such that* $(x_0,\cdots, x_s)\in\underset{\sim}{U}\in\underline{\underline{F}}$, $(x_s,\cdots, x_t)\in\underset{\sim}{V}\in\underline{\underline{F}}$, and $\underset{\sim}{U}\underset{\sim}{V} \subset \underset{\sim}{W}$.

For a simple but interesting example, suppose that a set of special edges of G is given, and for $0\le k\le \ell$ let $\underset{\sim}{W}(k)$ denote the set of all walks $(x_0,\cdots,x_t)$ such that (x_{i-1}, x_i) is special for at most k values of i. Let $\underline{\underline{F}} = \{\underset{\sim}{W}(k) = 0\le k\le \ell\}$. Note that $\underset{\sim}{W}(i)\underset{\sim}{W}(j) \subset \underset{\sim}{W}(k)$ when $i+j\le k$.

The *edges* of a walk $W = (x_0,\cdots, x_t)$ are $(x_0, x_1),\cdots, (x_{t-1}, x_t)$. A path P is *associated* with W if P also starts at x_0 and ends at x_t, P is a subsequence of W, and each edge of P is an edge of W. It is assumed $\underline{\underline{F}}$, L and λ are interrelated as follows:

(4) *if* $W\in\underset{\sim}{W}\in\underline{\underline{F}}$ *and* P *is a path associated with* W *then* $P\in\underset{\sim}{W}$ *and* $L_\lambda(P)\le L_\lambda(W)$.

Since there are only finitely many paths in G, a consequence of (4) is:

(5) *For each* $x, y\in N$ *and* $\underset{\sim}{Z}\in\underline{\underline{F}}$, *either there is no* $\underset{\sim}{Z}$*-walk from* x *to* y *or there is a* $\underset{\sim}{Z}$*-path which is a shortest* $\underset{\sim}{Z}$*-walk from* x *to* y.

As is shown in the last section of the paper, a number of problems on shortest restricted paths can be formulated and efficiently solved in terms of a function L and a family $\underline{\underline{F}}$ of sets of walks satisfying conditions (1) - (4).

The Algorithms.

The extended version of Floyd's algorithm (EVFA) starts with the $n \times n$ matrix λ of edge-lengths of G, procedures for computing $L(\alpha)$ and $L(\alpha, \beta)$ for all $\alpha, \beta \in R^*$, and a suitable representation of the family F of sets of walks. Also required is a set T of triples (U, V, W) of members of F such that:

(6) $UV \subset W$ <u>for each</u> $(U, V, W) \in T$;

(7) <u>if</u> $0<s<t$ <u>and</u> $(x_0, \cdots, x_t) \in W \in F$ <u>then there exist</u> U <u>and</u> V <u>such that</u> $(x_0, \cdots, x_s) \in U$, $(x_s, \cdots, x_t) \in V$, <u>and</u> $(U, V, W) \in T$.

Of course T may be taken as the set of all triples (U, V, W) of members of F such that $UV \subset W$, but it is most efficient to have T as small as possible subject to (6) and (7). (In the example following (3) in the preceding section, it would be best to let $T = \{(W(i), W(j), W(k)): i+j = k \leq \ell\}$ rather than using $i+j \leq k$.)

For each $Z \in F$ the algorithm outputs four $n \times n$ matrices: an R^*-valued S_Z, an integer-valued M_Z, an F-valued U_Z and an F-valued V_Z. At the time of output these satisfy the following conditions for all $x, y \in N$:

(8) <u>if there is no</u> Z<u>-path from</u> x <u>to</u> y <u>then</u> $S_Z[x, y] = \infty$ <u>and</u> $M_Z[x, y] = -1$;

(9) <u>if there is a</u> Z<u>-path from</u> x <u>to</u> y <u>then</u>

(a) $S_Z[x, y]$ <u>is the (possibly</u> ∞<u>) length of a shortest</u> Z<u>-path from</u> x <u>to</u> y;

(b) $M_Z[x, y] = 0$ <u>if</u> (x, y) <u>is such a shortest path</u>; <u>otherwise</u>, $M_Z[x, y]$ <u>is the index</u> m <u>of an intermediate node on such a shortest path, and the path itself is formed from a shortest</u> $U_Z[x, y]$<u>-path from</u> x <u>to</u> m <u>followed by a shortest</u> $V_Z[x, y]$<u>-path from</u> m <u>to</u> y.

Using the output M, U and V of EVFA the path-tracing algorithm (PTA) actually finds the shortest paths.

EVFA: EXTENDED VERSION OF FLOYD'S ALGORITHM

```
begin
  for i ← 1 until n do
    for j ← 1 until n do
      for each W∈F do
        if (i, j)∈W then begin
                           S_W[i, j] ← L(λ(i, j));
                           M_W[i, j] ← 0
                         end
                    else begin
                           S_W[i, j] ← ∞;
                           M_W[i, j] ← -1
                         end of initialization;
  for k ← 1 until n do
    for i ← 1 until n do
      for j ← 1 until n do
        for each (U, V, W)∈T do
          if (L(S_U[i, k], S_V[k, j]) < S_W[i, j])
            or (M_W[i, j] = -1 and M_U[i, k] ≠ -1 and M_V[k, j] ≠ -1)
              then begin
                     S_W[i, j] ← L(S_U[i, k], S_V[k, j]);
                     M_W[i, j] ← k;
                     U_W[i, j] ← U;
                     V_W[i, j] ← V
                   end of main loop;
```

```
for each W∈F do begin
                    print S_W, M_W, U_W and V_W
                end
end
```

In the path-tracing algorithm, STACK's members are alternately node-indices and members of $\underline{\underline{F}}$. (It is often convenient in practice to represent the members of $\underline{\underline{F}}$ by negative integers.) When a shortest $\underset{\sim}{Z}$-path from x to y is desired, STACK is initialized as $(y, \underset{\sim}{Z}, x)$. As STACK is processed, node-indices are added to PATH, which emerges as a shortest $\underset{\sim}{Z}$-path from x to y. (PTA has no output when there is no $\underset{\sim}{Z}$-path from x to y.)

PTA: PATH-TRACING ALGORITHM

```
begin
  if M_Z[x, y] = -1 then goto NONE;
  STACK[1] ← y; STACK[2] ← Z; STACK[3] ← x;
  s ← 3; p ← 0;
  while s ≥ 3 do
    begin
      i ← STACK[s];
      W ← STACK[s-1];
      j ← STACK[s-2];
      m ← M_W[i, j];
      if m = 0 then begin
                      p ← p+1;
                      PATH[p] ← i;
                      s ← s-2
```

```
                end
            else begin
                    STACK[s-1] ← V_W[i, j];
                    STACK[s] ← m;
                    STACK[s+1] ← U_W[i, j];
                    s ← s+2;
                    STACK[s] ← i
                end
    end of loop;
  PATH[p+1] ← STACK[1];
  print PATH;
  NONE:
end
```

In describing the efficiency of EVFA and PTA we use the uniform cost criterion in the RAM model of random access computation [AHU]. Initialization of EVFA requires time $O(|\underline{\underline{F}}|n^2)$ and each passage through the main loop requires time $O(|\underline{\underline{T}}|n^2)$, so the overall time-complexity of EVFA is

$$O(|\underline{\underline{F}}|n^2 + |\underline{\underline{T}}|n^3).$$

This counts each evaluation of $L(\alpha)$ or $L(\alpha, \beta)$ as a single step.

For each choice of $x, y, \underset{\sim}{Z}$, PTA requires time $O(n)$ to find a shortest $\underset{\sim}{Z}$-path from x to y. Thus, starting from the output of EVFA, time $O(|\underline{\underline{F}}|n^2)$ is required to find, for all $x, y \in N$ and $\underset{\sim}{Z} \in \underline{\underline{F}}$, a shortest $\underset{\sim}{Z}$-path from x to y.

It remains to show that EVFA and PTA compute what is claimed for them. In the case of PTA, this follows from a routine inductive argument showing that at

the end of the initialization and also at the end of each passage through the loop there exists a shortest $\underset{\sim}{Z}$-path P from x to y such that the following three conditions are satisfied:

(a) s is odd; STACK[1] = y; if $p > 0$ then PATH[1] = x;

(b) PATH[1],$\cdots$, PATH[p], STACK[s], STACK[s-2],$\cdots$, STACK[1] is a subsequence of P;

(c) for each odd k with $3 \le k \le s$, the segment of P that joins i = STACK[k] to j = STACK[k-2] is a shortest STACK[k-1]-path from i to j.

We turn now to EVFA. For $\underset{\sim}{W} \in \underset{=}{F}$ and $0 \le k \le n$, let $\underset{\sim}{W}_k$ denote the set of all walks $(x_0, \cdots, x_t) \in \underset{\sim}{W}$ such that $x_s \le k$ when $0 < s < t$. Thus the end nodes of walks in $\underset{\sim}{W}_k$ are unrestricted but all intermediate nodes are in $\{1, \cdots, k\}$. Let $\underset{=}{F}_k = \{\underset{\sim}{W}_k : \underset{\sim}{W} \in \underset{=}{F}\}$. Since $\underset{\sim}{W}_n = \underset{\sim}{W}$ for all $\underset{\sim}{W} \in \underset{=}{F}$, it suffices to prove the following for $0 \le k \le n$:

(10_k) <u>After the k^{th} passage through the main loop of</u> EVFA, <u>conditions</u> (8) <u>and</u> (9) <u>are satisfied for all</u> $x, y \in N$ <u>and</u> $\underset{\sim}{Z} \in \underset{=}{F}_k$.

The proof is by induction on k. Here initialization is regarded as the 0^{th} passage through the main loop and assertion (10_0) is obvious because $\underset{\sim}{Z}_0$ is merely the set of all edges (paths (x_0, x_1)) in $\underset{\sim}{Z}$.

From the argument below it follows that (10_k) holds for all k regardless of the order in which i, j and $(\underset{\sim}{U}, \underset{\sim}{V}, \underset{\sim}{W})$ appear in the main loop. In particular, the main loop could be written as

<u>for</u> $k \leftarrow 1$ <u>until</u> n <u>do</u>

  <u>for each</u> $(\underset{\sim}{U}, \underset{\sim}{V}, \underset{\sim}{W}) \in \underset{=}{T}$ <u>do</u>

    <u>for each</u> $(i, j) \in \{1, \cdots, n\} \times \{1, \cdots, n\}$ <u>do</u> . . . .

That is convenient for programming in some languages with special array-handling capabilities, such as APL.

Now suppose, with $0 \le k \le n$, that (10_{k-1}) holds, and consider the k^{th} passage through the main loop. We note first that if i or j is k then there do not exist $\underset{\sim}{U}, \underset{\sim}{V} \in \underset{=}{F}$ such that $\underset{\sim}{U}\underset{\sim}{V} \subset \underset{\sim}{W}$ and

(d) $M_{\underset{\sim}{W}}[i, j] = -1$ but $M_{\underset{\sim}{U}}[i, k] \neq -1 \neq M_{\underset{\sim}{V}}[k, j]$ or

(e) $L(S_{\underset{\sim}{U}}[i, k], S_{\underset{\sim}{V}}[k, j]) < S_{\underset{\sim}{W}}[i, j]$.

Suppose, for example, that $j = k$. If (d) holds there is no $\underset{\sim}{W}_{k-1}$-path from i to k but there is a $\underset{\sim}{U}_{k-1}$-path U from i to k and there is a $\underset{\sim}{V}_{k-1}$-path V from k to k. But then UV is a $\underset{\sim}{W}_k$-walk from i to k and by condition (4) there is an associated $\underset{\sim}{W}_k$-path P from i to k. Plainly P is, in fact, a $\underset{\sim}{W}_{k-1}$-path, and that is a contradiction. A similar contradiction is derived from (e), using conditions (1) - (4) and the inductive hypothesis. It follows that the k^{th} rows and k^{th} columns of $S_{\underset{\sim}{W}}$, $M_{\underset{\sim}{W}}$, $U_{\underset{\sim}{W}}$ and $V_{\underset{\sim}{W}}$ are unchanged by the k^{th} passage through the main loop. Hence (10_k) holds for all $\underset{\sim}{Z} \in \underset{=}{F}_k$ and $x, y \in N$ with $x = k$ or $y = k$. The case in which $x \neq k \neq y$ remains.

Supposing, still, that $0 < k \le n$ and (10_{k-1}) holds, consider $\underset{\sim}{W} \in \underset{=}{F}$ and $i, j \in N$ with $i \neq k \neq j$. We discuss only the case in which (e) holds at some time during the k^{th} passage, for the other cases (described in terms of (d) and (e)) are similar. Let

$$\mu = \min\{L(S_{\underset{\sim}{U}}[i, k], S_{\underset{\sim}{V}}[k, j]): (\underset{\sim}{U}, \underset{\sim}{V}, \underset{\sim}{W}) \in \underset{=}{T}\},$$

the minimum during the k^{th} passage, and let $(\underset{\sim}{U}', \underset{\sim}{V}')$ be the first pair $(\underset{\sim}{U}, \underset{\sim}{V})$ for which the minimum is attained. Then at the end of the k^{th} passage,

$$S_{\underset{\sim}{W}}[i,j] = \mu;\ M_{\underset{\sim}{W}}[i, j] = k,\ U_{\underset{\sim}{W}}[i, j] = \underset{\sim}{U}',\ V_{\underset{\sim}{W}}[i, j] = \underset{\sim}{V}'\ .$$

Let U be a shortest $\underset{\sim}{U}'_{k-1}$-path from i to k and let V be a shortest $\underset{\sim}{V}'_{k-1}$-path from k to j, whence $L_\lambda(UV) = \mu$. Then $UV \in \underset{\sim}{U}'\underset{\sim}{V}' \subset \underset{\sim}{W}$ and hence UV is

a $\underset{\sim}{W}_k$-walk from i to j. For (10_k) it suffices to show UV is a shortest $\underset{\sim}{W}_k$-path from i to j. Consider an arbitrary shortest $\underset{\sim}{W}_k$-walk W from i to j and an arbitrary associated path $P = (x_o, \cdots, x_t)$. Then $P \in \underset{\sim}{W}_k$ and $L_\lambda(P) \le L_\lambda(W)$, whence of course

$$L_\lambda(P) = L_\lambda(W) \le L_\lambda(UV).$$

The node k appears in P for otherwise it is true at the end of the $(k-1)^{th}$ passage that $S_{\underset{\sim}{W}}[i, j] = L_\lambda(P)$ and then (e) never holds during the k^{th} passage, contrary to hypothesis. With $x_s = k$ there exist $\underset{\sim}{U}$ and $\underset{\sim}{V}$ such that $(x_o, \cdots, x_s) \in \underset{\sim}{U}$, $(x_s, \cdots, x_t) \in \underset{\sim}{V}$ and $(\underset{\sim}{U}, \underset{\sim}{V}, \underset{\sim}{W}) \in \underset{=}{T}$. But then

$$L_\lambda(UV) \le L(S_{\underset{\sim}{U}}[i, k], S_{\underset{\sim}{V}}[k, j] \le L(L_\lambda(x_o, \cdots, x_s), L_\lambda(x_s, \cdots, x_t)) = L(P),$$

whence $L_\lambda(UV) = L_\lambda(W)$ and UV is a shortest $\underset{\sim}{W}_k$-walk from i to j. If UV is not a path it has an associated path that misses k, and that was shown to be impossible.

The Applications.

Though conditions (1)-(4) suffice for the validity of EVFA, some additional conditions aid in verifying condition (4) for specific applications. The function L is said to be nice if in addition to (1) and (2) it satisfies the following two conditions:

(11) each point of rng L is fixed under L; that is, $L(\alpha_1,\cdots,\alpha_t) = L(L(\alpha_1,\cdots,\alpha_t))$;

(12) if $\beta \geq 0$ then $L(\alpha,\beta) \geq L(\alpha) \leq L(\beta,\alpha)$ for all $\alpha \in$ rng L.

Note that each of L_p, L_∞ and L^p is nice.

If W is a walk $(x_0,\cdots,x_t)$ and the proper segment $(x_r,\cdots,x_s)$ of W is a circuit C then C is called an intermediate circuit of W and W_{rs} denotes the walk that remains when all of C but x_r or x_s is removed from W. More precisely, when $0 < r$ W_{rs} is the walk $(x_0,\cdots,x_r,x_{s+1},\cdots,x_t)$, $(x_0,\cdots,x_r,x_{s+1})$ or $(x_0,\cdots,x_r)$ according as $s+1<t$, $s+1 = t$ or $s = t$, and when $s < t$ W_{rs} is the walk $(x_0,\cdots,x_{r-1},x_s,\cdots,x_t)$, $(x_{r-1},x_s,\cdots,x_t)$ or $(x_s,\cdots,x_t)$ according as $r-1>0$, $r-1 = 0$ or $r = 0$.

Note that a walk is a path if and only if it has no intermediate circuit. Hence condition (4) can be deduced from repeated application of the following condition:

(13) if $W = (x_0,\cdots,x_t) \in \underset{\sim}{W} \in \underset{=}{F}$ and $(x_r,\cdots,x_s)$ is an intermediate circuit of W then $W_{rs} \in \underset{\sim}{W}$ and $L_\lambda(W_{rs}) \leq L_\lambda(W)$.

Note that the inequality of (13) always holds when $L = L_\infty$. In other cases it can often be deduced from the following result.

(14) If L is nice, $C = (x_r,\cdots,x_s)$ is an intermediate circuit of a walk $W = (x_0,\cdots,x_t)$ and $L_\lambda(C) \geq 0$ then $L_\lambda(W_{rs}) \leq L_\lambda(W)$.

To prove (14), note that $W = CW_{rs}$ if $r = 0$, $W = W_{rs}C$ if $s = t$, and if $0<r<s<t$ there are walks U and V such that $W = UCV$ and $W_{rs} = UV$. We consider

only the third case for the others are similar to it. With $L_\lambda(C) \geq 0$, it follows from (11), (12) and (1) that

$$L_\lambda(U) = L(L_\lambda(U)) \leq L(L_\lambda(U), L_\lambda(C)) = L_\lambda(UC),$$

then from (1) and (2) that

$$L_\lambda(W_{rs}) = L_\lambda(UV) = L(L_\lambda(U), L_\lambda(V)) \leq L(L_\lambda(UC), L_\lambda(V)) = L_\lambda(UCV) = L_\lambda(W).$$

Below are some illustrative problems on shortest restricted paths. In each case, "find shortest paths" means that for each $x, y \in N$, either a shortest path from x to y (among those satisfying the indicated restrictions) must be found or it must be concluded that no path from x to y satisfies the restrictions.

(A) A set of edges is given. For each $k \leq \ell$, find shortest paths that use at most k of the special edges.

(B) A set of nodes is given. For each $k \leq \ell$, find shortest paths that use at most k of the special nodes.

(C) A sequence of s sets is given, each consisting of nodes or edges or a mixture. For each choice of $(k_1, \cdots, k_s)$ with $k_r \leq \ell_r$ for all r, find shortest paths that use (for all r) at most k_r of the elements of the r^{th} set.

(D) In addition to the R^*-valued edge-lengths $\lambda(x, y)$, integer edge-lengths $\pi(x,y) \geq 0$ are given. Each walk has its usual length L_λ and also a length I_π where I is L_1. An integer $\ell \geq 0$ is given. For each $k \leq \ell$, find L_λ-shortest paths P subject to the restriction that $I_\lambda(P) \leq k$.

(E) The nodes of G are partitioned into two disjoint sets A and B, and an integer $\ell \geq 0$ is given. For each $k \leq \ell$, find shortest paths that oscillate at most k times between A and B.

(F) A subgraph H of G and an integer $\ell \geq 0$ are given. For each $k \leq \ell$, find shortest paths P for which $P \cap H$ has at most k components.

(G) A set M of edges of G is given. Find shortest M-alternating paths.

As can be seen by reference to conditions (1)-(4) and (13)-(14), *the discussions of* (A)-(G) *below are valid* (*that is*, EVFA *can be applied for the stated purpose*) *if* L *is* L_∞ *and also if* L *is nice and* $L_\lambda(C) \geq 0$ *for each circuit* C *intermediate to a walk belonging to a member of* $\underline{\underline{F}}$, *where* $\underline{\underline{F}}$ *is the family of sets of walks used for the particular problem*. (*Problem* (G) *requires an additional condition*, *stated later*.)

(A) This problem, which was mentioned earlier, is straightforward. For $0 \leq k \leq \ell$, let $\underset{\sim}{W}(k)$ denote the set of all walks $(x_0, \cdots, x_t)$ such that the edge (x_{i-1}, x_i) is special for at most k values of i. Let $\underline{\underline{F}} = \{\underset{\sim}{W}(k) \colon 0 \leq k \leq \ell\}$. Let $\underline{\underline{T}} = \bigcup_{0=k}^{\ell} \underline{\underline{T}}_k$ where

$$\underline{\underline{T}}_k = \{(\underset{\sim}{W}(i), \underset{\sim}{W}(j), \underset{\sim}{W}(k)) \colon i+j = k\}.$$

Then $|\underline{\underline{T}}_k| = k+1$ and $|\underline{\underline{T}}| = (\ell+1)(\ell+2)/2$. The overall time-complexity of EFWA for this problem is $O(\ell^2 n^3)$.

(B) This problem is similar to (A), but it is included to illustrate the way in which the end behavior of walks must sometimes be considered in constructing $\underline{\underline{F}}$ and $\underline{\underline{T}}$ for the application of EFWA. For $0 \leq k \leq \ell$ let $\underset{\sim}{W}_{--}(k)$ $\langle$resp. $\underset{\sim}{W}_{-+}(k)$, $\underset{\sim}{W}_{+-}(k)$, $\underset{\sim}{W}_{++}(k)\rangle$ denote the set of all walks $(x_0, \cdots, x_t)$ such that the node x_i is special for at most k values of i with $0 < i < t$ and, in addition, neither x_0 x_t is special $\langle$resp. x_t is special but x_0 is not, x_0 is special but x_t is not, both x_0 and x_t are special$\rangle$. Let $\underline{\underline{F}}$ consist of the sets $\underset{\sim}{W}_{--}(k)$ for $k \leq \ell$, the sets $\underset{\sim}{W}_{-+}(k)$ and $\underset{\sim}{W}_{+-}(k)$ for $k \leq \ell-1$, and the sets $\underset{\sim}{W}_{++}(k)$ for

$k \le \ell - 2$. Let $\underline{\underline{T}} = \bigcup_{o=k}^{\ell} T_k$, where $\underline{\underline{T}}_k$ consists of the triples

$$(W_{--}(i),\ W_{--}(j),\ W_{--}(k)) \qquad \text{for } i+j = k,$$

$$(W_{--}(i),\ W_{-+}(j),\ W_{-+}(k-1)), \qquad (W_{+-}(i),\ W_{--}(j),\ W_{+-}(k-1))$$

$$\text{and} \qquad (W_{-+}(i),\ W_{+-}(j),\ W_{--}(k)) \qquad \text{for } i+j = k-1,$$

$$(W_{-+}(i),\ W_{++}(j),\ W_{-+}(k-1)), \qquad (W_{++}(i),\ W_{+-}(j),\ W_{+-}(k-1))$$

$$\text{and} \qquad (W_{+-}(i),\ W_{-+}(j),\ W_{++}(k-2)) \qquad \text{for } i+j = k-2,$$

$$\text{and} \qquad (W_{++}(i),\ W_{++}(j),\ W_{++}(k-2)) \qquad \text{for } i+j = k-3.$$

Again, $|\underline{\underline{T}}|$ is $O(\ell^2)$ and the complexity of EVFA is $O(\ell^2 n^3)$.

(C) This is included to illustrate the application of EVFA when the desired paths are subject to several restrictions. In order to avoid a notational morass, only the case of sets of edges is discussed. For $k_1 \le \ell_1, \cdots, k_s \le \ell_s$, let $W(k_1, \cdots, k_s)$ denote the set of all walks $(x_0, \cdots, x_t)$ such that, for $1 \le r \le s$, (x_{i-1}, x_i) belongs to the r^{th} set of edges for at most k_r values of i. Let $\underline{\underline{T}}$ consist of all triples

$$(W(i_1, \cdots, i_s),\ W(j_1, \cdots, j_s),\ W(k_1, \cdots, k_s))$$

such that for $1 \le r \le s$, $i_r + j_r = k_r \le \ell_r$. Then

$$|\underline{\underline{T}}| = \sum_{k_1=0}^{\ell_1} \sum_{k_2=0}^{\ell_2} \cdots \sum_{k_s=0}^{\ell_s} ((k_1+1)(k_2+1) \cdots (k_s+1)) = 2^{-s} \prod_{r=1}^{s} (\ell_r+1)(\ell_r+2)$$

and the complexity of EVFA is

$$O(2^{-s}(\ell_1 \ell_2 \cdots \ell_s)^2 n^3).$$

(D) This may be regarded as the integer-weighted version of a problem of which (A) is the cardinality-weighted version. Similar extensions are available for the

other problems considered here. For $0 \le k \le \ell$, let $\underset{\sim}{W}(k)$ be the set of all walks W for which $I_\pi(W) \le k$. Define $\underline{\underline{F}}$ and $\underline{\underline{T}}$ in the obvious ways. The complexity of EVFA is $O(\ell^2 n^3)$. For a closely related treatment of this problem and of (A), see the discussion of the Bellman-Ford method in [L, pp. 74-75, 92-93]. EVFA is similar to the Bellman-Ford method but is more general. Roughly speaking, it amounts to replacing the additive semigroup $\{0,1,2,\ldots\}$ of Bellman-Ford by an arbitrary semigroup.

(E) This is a special case of a more general problem, which may be formulated as follows: A function ϕ is defined on a set of nodes and edges of G, with $\operatorname{rng} \phi \subset \{1,\cdots,m\}$, and an integer $\ell \ge 0$ is given. For each $k \le \ell$, find shortest paths along which ϕ has at most k relative extrema.

As the term is used here, a <u>relative extremum</u> of a real sequence $(\alpha_0,\cdots,\alpha_u)$ is an ordered pair (r, s) such that $0<r\le s<u$ and

$$\alpha_{r-1} < \alpha_r = \cdots = \alpha_s > \alpha_{s+1} \quad \text{or} \quad \alpha_{r-1} > \alpha_r = \cdots = \alpha_s < \alpha_{s+1}.$$

For a walk $W = (x_0,\cdots,x_t)$, let

$$\exp W = (x_0, (x_0, x_1), x_1,\cdots,(x_{t-1},x_t), x_t),$$

the expanded version of W in which nodes and edges alternate. Let W_ϕ denote the sequence of ϕ-values corresponding to the elements of $\exp W$ that belong to $\operatorname{dmn} \phi$, and let $\rho_\phi(W)$ denote the number of relative extrema of W_ϕ. The general problem is to find shortest paths P for which $\rho_\phi(P) \le k$. Problem (E) is the special case in which

(*) $m = 2$ and $\operatorname{dmn} \phi = N = A \cup B$, with $\phi = 1$ on A and $\phi = 2$ on B.

As is shown below, this can be handled by EVFA. However, we do not know how to use EVFA efficiently for the general problem, or even for the following special cases:

$m = 3$ and dmn $\phi = N$;

$m = 2$ and dmn ϕ is a proper subset of N;

$m = 2$ and dmn ϕ is the set E of all edges of G.

In each case there is difficulty, even when L is L_1 and all values of the edge-length λ are positive, in constructing a suitable family $\underset{=}{F}$ satisfying conditions (3) and (4).

Now let us return to (E) in the formulation provided by (*), except that the condition dmn $\phi = N$ may be replaced by dmn $\phi \supset N$. For $0 \le k \le \ell$ and $u, v \in \{1,2\}$, let $\underset{\sim}{W}_{uv}(k)$ denote the set of all walks $W = (x_0, \cdots, x_t)$ such that $\phi(x_0) = u$, $\phi(k_t) = v$, and W_ϕ has at most k relative extrema (equivalently, W oscillates at most k times between A and B). Let

$$\underset{=}{F} = \{W_{u,v}(k) \colon u, v \in \{1, 2\}, k \le \ell\}$$

and let $\underset{=}{T}$ consist of all triples.

$(\underset{\sim}{W}_{uu}(i), \underset{\sim}{W}_{uu}(j), \underset{\sim}{W}_{uu}(k))$ for $u \in \{1,2\}$ and $(\{i,j\} = \{0,k\}$ or $(i>0<j$ and $i+j = k-1))$,

$(\underset{\sim}{W}_{uv}(i), \underset{\sim}{W}_{uu}(j), \underset{\sim}{W}_{uu}(k))$ for $\{u,v\} = \{1,2\}$ and $i+j = k-1$,

$(\underset{\sim}{W}_{uv}(i), \underset{\sim}{W}_{vv}(j), \underset{\sim}{W}_{uv}(k))$ for $\{u,v\} = \{1,2\}$ and $((i,j) = (k,0)$ or $(j>0$ and $i+j = k-1))$,

$(\underset{\sim}{W}_{uu}(i), \underset{\sim}{W}_{uv}(j), \underset{\sim}{W}_{uv}(k))$ for $\{u,v\} = \{1,2\}$ and $((i,j) = (0,k)$ or $(i>0$ and $i+j = k-1))$.

Then EVFA can be applied, solving problem (E) in time $O(\ell^2 n^3)$.

(F) Define ϕ on all nodes and edges of G, with $\phi = 1$ on nodes and edges of the graph H and $\phi = 2$ otherwise. With the $\underset{\sim}{W}_{uv}$ as in the preceding paragraph, the paths P for which $P \cap H$ has at most k components are precisely the paths in

$$\underset{\sim}{W}_{11}(2k-3) \cup \underset{\sim}{W}_{12}(2k-2) \cup \underset{\sim}{W}_{21}(2k-2) \cup \underset{\sim}{W}_{22}(2k-3).$$

Hence (F) can also be handled by EVFA in time $O(\ell^2 n^3)$.

(G) This problem is also discussed in a more general setting. With ϕ as in the discussion of (E) and with $m = 2$, let a walk W be called ϕ-alternating if the sequence W_ϕ alternates between 1 and 2. How can shortest ϕ-alternating paths be found? Problem (G) is the special case in which $\text{dmn}\ \phi = E$, $\phi = 1$ on M, and $\phi = 2$ on $E \sim M$.

When $\text{dmn}\ \phi \supset N$, EVFA can be applied by taking $\underset{=}{F} = \{\underset{\sim}{W}_{11}, \underset{\sim}{W}_{12}, \underset{\sim}{W}_{21}, \underset{\sim}{W}_{22}\}$, where $\underset{\sim}{W}_{uv}$ is the set of all ϕ-alternating walks W such that the sequence W_ϕ starts with u and ends with v. Then let $\underset{=}{T}$ consist of all triples $(\underset{\sim}{W}_{uv}, \underset{\sim}{W}_{vu}, \underset{\sim}{W}_{uu})$ and $(\underset{\sim}{W}_{uv}, \underset{\sim}{W}_{vv}, \underset{\sim}{W}_{uv})$ for all $u, v \in \{1,2\}$.

Now consider the case in which $\text{dmn}\ \phi = E$. Then each pair (i, j) forms a ϕ-alternating path (recall the standing hypothesis that G is the complete graph on N) but of course we are interested only in paths of finite length. Define $\underset{=}{F}$ by restricting the $\underset{\sim}{W}_{uv}$'s of the preceding paragraph to include only walks of finite length, and assume

($\dagger$) each alternating circuit of finite length has an even number of edges.

Then problem (F) can be handled by EVFA with $\underset{=}{T}$ consisting of all triples $(\underset{\sim}{W}_{up}, \underset{\sim}{W}_{qv}, \underset{\sim}{W}_{uv})$ for $u, v \in \{0,1\}$ and $\{p,q\} = \{0,1\}$. However, this approach may fail when ($\dagger$) fails for then a path associated with an alternating walk need not be an alternating path and thus condition (4) may fail. For example, consider Fig. 1 and note that (x_2, X_3, X_4, x_5) is an alternating circuit according to our definition, where the solid edges are those in M.

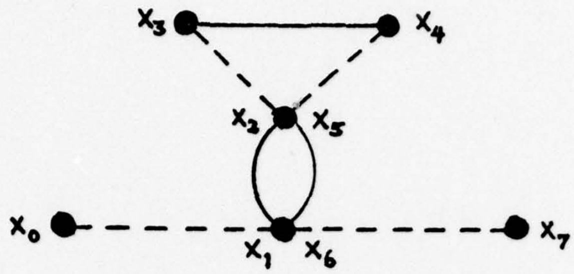


Fig. 1: The walk $(x_0,\cdots, x_7)$ is alternating but the associated path (x_0,x_1,x_7) is not.

Brown [B] suggests a method for finding shortest M-alternating paths in a directed graph $D = (N, E)$ (no longer assumed complete). Another directed graph D^* is constructed, having two nodes x' and x'' for each node x of D, and the edges of D^* are obtained as follows for each edge (x, y) of D with length $\lambda(x,y)<\infty$.

when $(x,y) \in M$, (x',y'') is an edge of D^* with length $\lambda(x, y)$;

when $(x,y) \notin M$, (x'', y') is an edge of D^* with length $\lambda(x, y)$.

It is claimed [B] there is a natural one-to-one correspondence between alternating paths in D and ordinary paths in D^*. Thus the problem of finding shortest alternating paths in D is equivalent to the problem of finding shortest ordinary paths in D^*. The claim is correct when (+) holds but not in general, as can be seen from Fig. 1. For example, if D is the graph of Fig. 1 then the path $(x_0'',x_1',x_2'',x_3',x_4'',x_2',x_1'',x_4')$ in D^* corresponds to the walk $(x_0,x_1,x_2,x_3,x_4,x_2,x_1,x_7)$ in D.

In general, Brown's construction does produce a one-to-one correspondence between the walks in D^* and the alternating walks in D. If D has no negative alternating circuit then D^* has no negative circuit and the Floyd-Warshall algorithm can be applied to find shortest paths (=shortest walks) in D^* and hence alternating walks in D. The latter may or may not be paths. For general graphs, even when $L = L_1$ and may or may not be paths. For general graphs, even when $L = L_1$ and $\lambda > 0$, we

do not know how to apply EFVA directly to find shortest alternating paths. However, the more complicated "blossom" methods of Edmonds [E1][E2][L] will apparently apply to this problem.

We close with a query. For $0 \le \ell \le m < n$ and for $\rho \in \{\le, =, \ge\}$ let $\underline{\underline{P}}_0(\ell,m,n,\rho)$ $\langle$resp. $\underline{\underline{P}}_1(\ell,m,n,\rho)\rangle$ denote the following problem:

A complete graph G is given, with n nodes and positive edge-lengths. In addition, a set of m special nodes $\langle$resp. edges$\rangle$ of G is given. Find shortest paths P in G such that the number of special nodes $\langle$resp. edges$\rangle$ used by P is in the relation ρ to ℓ.

We have seen here that the extended version of Floyd's algorithm solves $\underline{\underline{P}}_0(\ell,m,n,\le)$ and $\underline{\underline{P}}_1(\ell,m,n,\le)$ in time $O(\ell^2 n^3)$. By contrast, the problem $\underline{\underline{P}}_0(n,n,n,=)$ is essentially the traveling salesman problem and hence is NP-complete [K1][K2][AHU]. What else of interest can be said about the computational complexity of $\underline{\underline{P}}_0(\ell,m,n,\rho)$ and $\underline{\underline{P}}(\ell,m,n,\rho)$ for $\rho \in \{=, \ge\}$?

References

[AHU] A. V. Aho, J. E. Hopcroft and J. D. Ullman, The Design and Analysis of Computer Algorithms. 1974. Addison-Wesley, Reading, Mass.

[B] J. R. Brown, Shortest alternating path algorithms, Networks 4(1976) 311-334.

[E1] J. Edmonds, Paths, trees and flowers, Can. J. Math. 17(1965) 449-467.

[E2] J. Edmonds, Maximum matching and a polyhedron with 0, 1 vertices, J. Res. Nat. Bur. Standards 69B(1965) 125-130.

[F] R. W. Floyd, Algorithm 97, Shortest path. Comm. Assoc. Comp. Mach. 5(1962)345.

[H1] T. C. Hu, The maximum capacity route problem. Operations Res. 9(1961)898-900.

[H2] T. C. Hu, Integer Programming and Network Flows. 1969. Addison-Wesley, Reading, Mass.

[K1] R. M. Karp, Reducibility among combinatorial problems. Complexity of Computer Computations (Proc. Sympos. IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y., 1972), pp. 85-103. 1972. Plenum, New York.

[K2] R. M. Karp, On the computational complexity of combinatorial problems. Networks 5(1975)45-68.

[L] E. L. Lawler, Combinatorial Optimization: Networks and Matroids. 1976. Holt, Rinehart and Winston, New York.

[R] B. Roy, Transitivité et connexité, C. R. Acad. Sci. Paris 249(1959) 216-218.

[W] S. Marshall, A theorem on Boolean matrices, J. Assoc. Comp. Mach. 9(1962) 11-12.

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) University of Washington	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

Use of Floyd's algorithm to find shortest restricted paths

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Technical

5. AUTHOR(S) (First name, middle initial, last name)

Victor Klee and David Larman

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
September, 1977	21	12

8a. CONTRACT OR GRANT NO. N00014-67-A-0103-0003	9a. ORIGINATOR'S REPORT NUMBER(S) Technical Report No. 62
b. PROJECT NO. NR044 353 c. d.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

Releasable without limitations on dissemination

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

In a directed network with no negative circuit, Floyd's algorithm finds, for each pair of nodes x and y, a shortest path from x to y. Here the procedure is extended to minimize more general length functions over sets of paths that are restricted in various ways.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6801

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
arc length						
directed network						
Floyd's algorithm						
network						
negative circuit						
restricted path						
shortest chain						
shortest path						
walk						